

## **METHOD OF REDIRECTING CLIENT REQUESTS TO WEB SERVICES**

### **Technical field**

**[0001]** The invention relates to the web services provided to the clients of the Internet network and in particular relates to a method of redirecting the client requests to the web service when its address has changed.

### **Background**

**[0002]** It is a common knowledge that it is possible to have access to more and more services on the Internet network. So the Internet users are interested in obtaining information about on-line services on the Internet and how to request these services. The on-line services are generally provided by application service providers, but there are also services which are domain-oriented like the request on life sciences.

**[0003]** Today, it is possible for a client to obtain a requested information by using the "Web Services" which is a specific business functionality exposed by a company, usually through an Internet connection, for the purpose of providing a way for a client, a company or a program to use the service. Web services are becoming a backbone for the electronic commerce and the exchange with the application service providers.

**[0004]** A service provider providing public web services on the Internet network has to describe the whole informations regarding the web service. To do that, he can provide a Web Service Definition Language (WSDL) file associated with the web service. A WSDL is useful to describe the interface (methods, parameters,...) and the runtime

(bindings, addresses,...) of the service. From the client point of view, a WSDL file contains all the information required to build the client structure. Normally, a client would use the WSDL file only at the build time and not at the run time.

**[0005]** In a normal production model, it is sure that a WSDL will evolve because of new functional or runtime requirements. The main difficulty when a service provider wants to update a web service and the associated WSDL file is the notification of all the web service clients. In fact, in case of a public web service, the service provider does not know the web service clients, and therefore, he cannot notify them of a change.

**[0006]** Some practices regarding changes in the interface of a web service are provided. Thus, a change in the interface should be always compatible with previous versions. For instance, new methods are allowed in a WSDL file, but the removal of these methods is forbidden.

**[0007]** However, there is no way for a service provider to notify all web service clients when the end point address of one of its public web service changes. The reasons are that he does not use a reverse proxy or a web service gateway and wants to move the web service from a first host to another more powerful host, or he wants to add a load-balancing solution in front of the web service, or he wants to add a web service gateway, a secure proxy in front of it, or he has to change the domain name.

**[0008]** An existing solution already exists for the HTTP protocol as transport protocol using 3xx HTTP status codes. A client browser receiving a 3xx HTTP response code knows that the location of the required document has changed and the browser automatically generates a new HTTP request to the location indicated in the 3xx HTTP

response code. But this solution cannot be applied for web services using Single Object Access Protocol (SOAP) as message exchange protocol for two reasons. Firstly, the codes 3xx are not automatically handled by runtimes that provide the binding between SOAP and HTTP. Secondly, the communication protocol for the SOAP messages is not only HTTP since the SOAP messages can also be sent on other transport protocols like SMTP, JMS,... and the 3xx HTTP status codes for redirection are not available on these protocols.

### **Summary of the invention**

**[0009]** Accordingly, the object of the invention is to achieve a method of redirecting a client of a web service to a new end point address in case of a runtime change.

**[0010]** The invention therefore relates to a method of redirecting a request for a web service in a data transmission network such as the Internet wherein, in response to a request forwarded by a host of a client to a web service provider, this one provides a Web Service Definition Language (WSDL) file based upon a message exchange protocol such as Simple Object Access Protocol (SOAP) on a transport protocol such as HTTP. This method comprises the steps of forwarding a first request from the client to an old address of the web service, responding to the client from the web service point associated with the old address by sending back a message wherein the header using the message exchange protocol contains the new address redirection of the requested web service, and forwarding a second request from the client to the new address of the web service.

**[0011]** According to another aspect, the invention relates to a system comprising a checker in the SOAP runtime of the service provider being adapted to check whether a request forwarded from the SOAP runtime of a client to the service provider has to be redirected to a new point address and to provide the new point address to which the request must be forwarded, this new point address being provided in a SOAP response message forwarded from the service provider at the old point address to the SOAP runtime of the client, and a checker in the SOAP runtime of the client adapted to check whether the SOAP response message contains the new point address and regenerate the same request and to forward it to the new point address.

### **Brief description of the drawings**

**[0012]** The above and other objects, features and advantages of the invention will be better understood by reading the following more particular description of the invention in conjunction with the accompanying drawings wherein:

- Fig. 1 is a schematic diagram representing the method according to the invention;
- Fig. 2 is a block-diagram representing the system according to the invention;
- Fig. 3 is a flow chart representing the steps of the method which is implemented in the SOAP runtime associated with the old web service address; and
- Fig. 4 is a flow chart representing the steps of the method which is implemented in the SOAP runtime of the client.

### **Detailed description of the invention**

**[0013]** In reference to FIG.1, it is assumed that a web service client 10 based upon the message exchange protocol SOAP wants to have access to web services provided by a service provider. In such a case, a first request is forwarded from the host 10 to the current web service address which is the old address point 12 since the user does not know that the address of the web service has been changed. The old point address 12 answers back to the host 10 by providing the redirection in the header of the response message as described later. Using the redirection contained in the SOAP message received from the old point address, the host 10 can then transmit a second request to the new web service address 14 in order to get a response to this request. Note that the host can be programmed to log the redirection in order to make further SOAP requests directly to the new point address.

**[0014]** An implementation of the method according to the invention is illustrated in FIG.2. The web service client station 10 includes normally two functionalities : a client application 18 and a SOAP runtime contains a checker 22 which is a component in charge of checking whether a received SOAP message contains a redirection in its header. If so, the checker regenerates the same SOAP request to the new point address provided in the header as mentioned above and optionally logs the redirection to allow code changes in the client application in order to direct the future requests to the right address. Note that the checker 22 is generally a routine within the runtime, but could be an independent component accessed by the client runtime each time it receives a SOAP message.

**[0015]** The web service provider 24 includes a first HTTP 26 server as usual and a SOAP runtime like the client one. The runtime is adapted to gain access to the web service such as web services WS 1, WS 2 and WS 3. As for the client, the SOAP runtime 28 of the service provider has at its disposal a checker 30 which can be a software part within the runtime or an independent component accessed when a SOAP request is received. Note that any other transport protocol than HTTP could be used such as SMTP. In the latter case, the server 26 would be a SMTP server.

**[0016]** Each time a SOAP request is received from a client 10, the checker 30 is adapted to check whether the address of the requested web service is the right address or an old address because the web service has moved to another machine or address and has become outdated. For this, the checker 30 has access to a component 32 knowing which are the web services that have been outdated and the new address if the requested service is outdated. Then, the checker is adapted to provide the new address to the runtime in order to have it included in the header of the SOAP message returned to the client.

**[0017]** The way to include the new address in the header of the SOAP message is to create a SOAP header tag that has the name "Redirect" and is included in the SOAP specification to ensure that all SOAP runtimes used by the clients and service providers will understand it. Nevertheless, the inclusion of the header in the SOAP specification is not mandatory. It could be a SOAP extension supported only by several runtimes which would provide a higher quality of service clients and service providers using these runtimes.

**[0018]** An example of such a header in a SOAP response message could be the following XML code:

```
<soapenv:Header>
```

```
  <soapenv:Redirect soapenv:mustUnderstand="1">
```

```
    Http://newhost.domain.com/services/PublicWebService</soapenv: Redirect>
```

```
</soapenv:Header>
```

**[0019]** The behavior of the server SOAP runtime server which provides the web services is illustrated in FIG.3. At the beginning, the SOAP runtime waits for a SOAP request (step 40). As already mentioned, the SOAP runtime has access to properties file, generally written in XML, that provides the list of the outdated web services. A new point address is associated with each of said outdated web services. Therefore, after getting a SOAP request, the checker retrieves the outdated web service in this list (step 42) and checks whether the web service requested by the client is a outdated web service (step 44). If the web service is present in this list of outdated web services, the SOAP runtime does not try to contact the web service but sends back a SOAP response with the SOAP redirection header (step 46). As already explained, this header contains the new point address provided by the properties file. The SOAP body is filled with the exact SOAP body received from the client in order to allow the client SOAP runtime running again the same SOAP request to the new point address.

**[0020]** If the web service is not present in the list of outdated web services, the SOAP runtime continues its normal work, that is forwarding the SOAP request to the

appropriate web service (step 48). In any case, the process is looped back to the starting step of waiting for a SOAP request (step 40).

**[0021]** The behavior of the client SOAP runtime is illustrated in FIG.4. At the beginning, the runtime waits for a SOAP response (step 50). Then the checker of the runtime checks whether this response contains a redirection header (step 52). If so, the SOAP runtime regenerates a new SOAP request to the new point address contained in the message header (step 54). The SOAP runtime also gets the SOAP body from the SOAP response and puts it in the new SOAP request. Optionally, the SOAP runtime can also log the redirection to allow further changes in the client code and therefore direct access to the new address without the need of requesting the redirection (step 56). If there is no redirection header in the SOAP response message, the SOAP runtime continues its normal work, that is forwarding the SOAP response to the client (step 58). In any case, the process is looped back to the starting step of waiting for a SOAP response (step 50).

**[0022]** Even though the Single Object Access Protocol (SOAP) is used in the preferred embodiment of the invention, it must be noted that any other message exchange protocol than SOAP could be used whatever the transport protocol being used.